
PRACTICAL GENETIC ALGORITHMS

RANDY L. HAUPT and SUE ELLEN HAUPT

Best Available Copy



A Wiley-Interscience Publication

JOHN WILEY & SONS, INC.

New York / Chichester / Weinheim / Brisbane / Singapore / Toronto

This text is printed on acid-free paper.

Copyright © 1998 by John Wiley & Sons, Inc. All rights reserved.

Published simultaneously in Canada.

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (508) 750-8400, fax (508) 750-4744. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 605 Third Avenue, New York, NY 10158-0012, (212) 850-6011, fax (212) 850-6008, E-Mail: PERMREQ@WILEY.COM.

Library of Congress Cataloging-in-Publication Data:

Haupt, Randy L.

Practical genetic algorithms / Randy L. Haupt & Sue Ellen Haupt.

p. cm.

"A Wiley-Interscience publication."

Includes bibliographical references and index.

ISBN 0-471-18873-5 (cloth : alk. paper)

1. Genetic algorithms. I. Haupt, S. E. II. Title.

QA402.5.H387 1998

519.7—dc21

97-13172

CIP

Printed in the United States of America

10 9 8 7 6 5 4 3 2

To find a peak, an optimization algorithm searches for the maximum cost. This analogy leads to the problem of finding the highest point in Rocky Mountain National Park. A three dimensional plot of a portion of the park (our search space) is shown in Figure 2.3 and a crude topographical map (128×128 points) with some of the highlights is shown in Figure 2.4. Locating the top of Long's Peak (14,255 ft above sea level) is the goal. Three other interesting features in the area include Storm Peak (13,326 ft), Mount Lady Washington (13,281 ft), and Chasm Lake (11,800 ft). Since there are many peaks in the area of interest, conventional optimization techniques have difficulty finding Long's Peak unless the starting point is in the immediate vicinity of the peak. In fact, all of the methods requiring a gradient of the cost function won't work with discrete data. The genetic algorithm has no problem!

2.2.1 Selecting the Parameters and the Cost Function

A cost function generates an output from a set of input parameters (a chromosome). The cost function may be a mathematical function, an experiment, or a game. The object is to modify the output in some desirable fashion by finding the appropriate values for the input parameters. We do

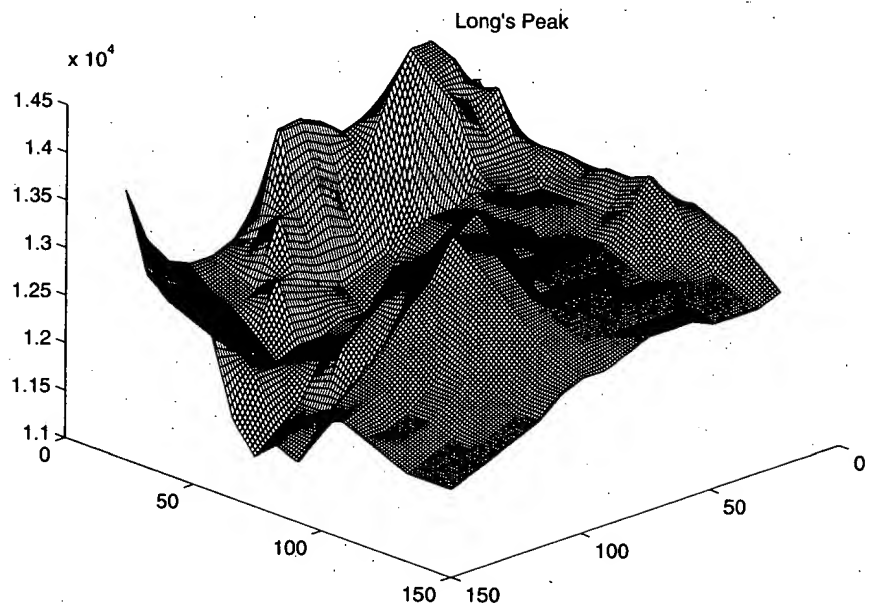


Figure 2.3 Three-dimensional view of the cost surface with a view of Long's Peak.

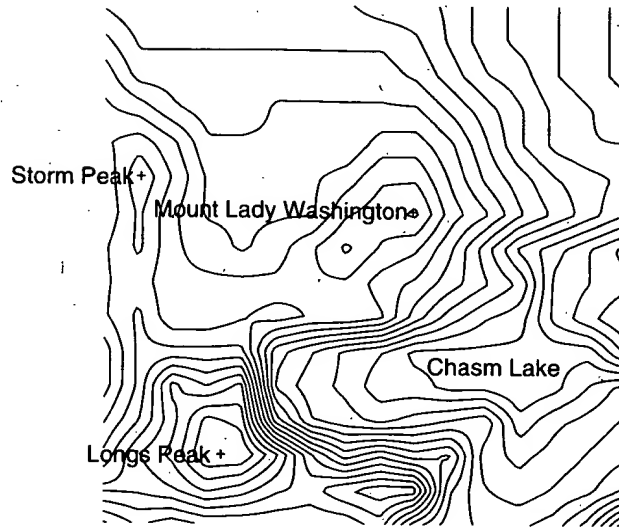


Figure 2.4 Contour plot or topographical map of the cost surface around Long's Peak.

this without thinking when filling a bathtub with water. The cost is the difference between the desired and actual temperatures of the water. The input parameters are how much the hot and cold spigots are turned. In this case, the cost function is the experimental results of feeling the resulting temperature. So we see that determining an appropriate cost function and deciding which parameters to use are intimately related.

The genetic algorithm begins by defining a chromosome or an array of parameter values to be optimized. If the chromosome has N_{par} parameters (an N_{par} -dimensional optimization problem) given by $p_1, p_2, \dots, p_{N_{par}}$, then the chromosome is written as an N_{par} element array.

$$chromosome = [p_1, p_2, p_3, \dots, p_{N_{par}}] \quad (2.1)$$

For instance, searching for the maximum elevation on a topographical map requires a cost function with input parameters of longitude (x) and latitude (y)

$$chromosome = [x, y] \quad (2.2)$$

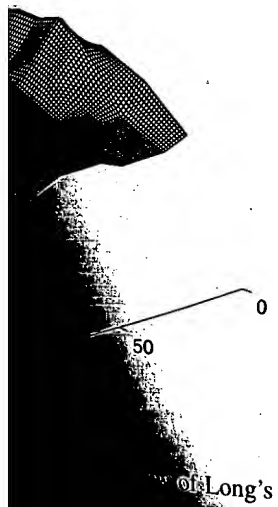
where $N_{par} = 2$. Each chromosome has a cost found by evaluating the cost function, f , at $p_1, p_2, \dots, p_{N_{par}}$:

$$cost = f(chromosome) = f(p_1, p_2, \dots, p_{N_{par}}) \quad (2.3)$$

es for the maximum cost.
e highest point in Rocky
ot of a portion of the park
crude topographical map
is shown in Figure 2.4.
ve sea level) is the goal.
e Storm Peak (13,326 ft),
Lake (11,800 ft). Since
onventional optimization
less the starting point is
of the methods requiring
discrete data. The genetic

Cost Function

of input parameters (a
mathematical function, an ex-
output in some desirable
input parameters. We do



Since we are trying to find the peak in Rocky Mountain National Park, the cost function is written as the negative of the elevation in order to put it into the form of a minimization algorithm:

$$f(x, y) = -\text{elevation} \quad (2.4)$$

Often, the cost function is quite complicated, as in maximizing the gas mileage of a car. The user must decide which parameters of the problem are most important. Too many parameters bog down the genetic algorithm. Important parameters for optimizing the gas mileage might include size of the car, size of the engine, and weight of the materials. Other parameters, such as paint color and type of headlights, have little or no impact on the car gas mileage and should not be included. Sometimes the correct number and choice of parameters comes from experience or trial optimization runs. Other times we have an analytical cost function, with the parameter being the variables of the function. A cost function defined by $f(w, x, y, z) = 2x + 3y + z/100000 + \sqrt{w}/9876$ with all parameters lying between 1 and 10 can be simplified to help the optimization algorithm. Since the w and z terms are extremely small in the region of interest, they can be discarded for most purposes. Thus, the four-dimensional cost function is adequately modeled with two parameters in the region of interest.

Most optimization problems require constraints or parameter bounds. Allowing the weight of the car to go to zero or letting the car width be 10 m are impractical parameter values. Unconstrained parameters can take any value. Constrained parameters come in three brands. First, hard limits in the form of $>$, $<$, \geq , and \leq can be imposed on the parameters. When a parameter exceeds a bound, then it is set equal to that bound. If x has limits of $0 \leq x \leq 10$, and the algorithm assigns $x = 11$, then x will be reassigned to the value of 10. Second, variables can be transformed into new variables that inherently include the constraints. If x has limits of $0 \leq x \leq 10$, then $x = 5 \sin y + 5$ is a transformation between the constrained variable x and the unconstrained variable y . Varying y for any value is the same as varying x within its bounds. This type of transformation changes a constrained optimization problem into an unconstrained optimization problem in a smooth manner. Finally, there may be a finite set of parameter values from which to choose, and all values lie within the region of interest. Such problems come in the form of selecting parts from a limited supply.

Dependent parameters present special problems for optimization algorithms, because varying one parameter also changes the value of the other parameter. For example, size and weight of the car are dependent. Increasing the size of the car will most likely increase the weight as well (unless

ky Mountain National Park, the elevation in order to put it

ation (2.4)

ated, as in maximizing the gas mileage might include size of the materials. Other parameters, have little or no impact on the. Sometimes the correct number of trial optimization runs. tion, with the parameter being defined by $f(w, x, y, z) =$ parameters lying between 1 and n algorithm. Since the w and z interest, they can be discarded. A cost function is adequately of interest.

straints or parameter bounds. For letting the car width be unconstrained parameters can take three brands. First, hard limits are placed on the parameters. When a limit is reached. If x has limits $x = 11$, then x will be reassigned. transformed into new variables. as limits of $0 \leq x \leq 10$, then the constrained variable x and its value is the same as varying. tion changes a constrained optimization problem in a set of parameter values from the region of interest. Such as from a limited supply.

blems for optimization algorithms changes the value of the other parameters. The car are dependent. Increase the weight as well (unless

some other factor, such as type of material, is also changed). Independent parameters, like Fourier series coefficients, do not interact with each other. If 10 coefficients are not enough to represent a function, then more can be added without having to recalculate the original ten.

In the genetic algorithm literature, parameter interaction is called *epistasis* (a biological term for gene interaction). When there is little to no epistasis, minimum seeking algorithms perform best. Genetic algorithms shine when the epistasis is medium to high, and pure random search algorithms are champions when epistasis is very high (Figure 2.5).

Defining the cost is a challenge. Should the cost only be related to one observable output, or should it be a combination of outputs? Let's look at the situation of the car design. Is letting $cost = \text{mi/gal}$ a good enough cost function for the design of a car? Assume the money needed to build the car (\$) and the gas mileage (M) are measures of the car design with parameters being weight (w) and volume (v) (We wouldn't buy this car!). An example cost function is

$$cost(w, v) = \$ + M \quad (2.5)$$

If the cost is about \$10,000 and the mileage 50 mi/gal, then the cost function is skewed toward the monetary cost of the vehicle. If both monetary cost and mileage are equally important, then the cost can be normalized as

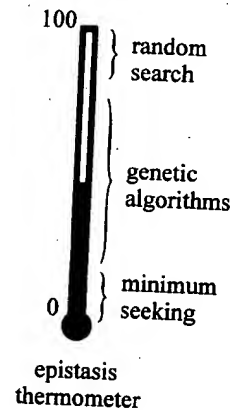


Figure 2.5 This graph of an epistasis thermometer shows that minimum seeking algorithms work best for low epistasis, while random algorithms work best for very high epistasis. Genetic algorithms work best in a wide range of medium to high epistasis.

follows:

$$cost = \frac{\$ - \$_{lo}}{\$_{hi} - \$_{lo}} + \frac{M - M_{lo}}{M_{hi} - M_{lo}} \quad (2.6)$$

where the *hi* and *lo* subscripts indicate the maximum and minimum values, respectively. This cost lies between 0 and 2, with the *lo* and *hi* subscripts representing the low and high values of the parameters. If one portion of the cost is more important, then it can be appropriately weighted ($0 < wt < 1$):

$$cost = wt \frac{\$ - \$_{lo}}{\$_{hi} - \$_{lo}} + (1 - wt) \frac{M - M_{lo}}{M_{hi} - M_{lo}} \quad (2.7)$$

The genetic algorithm is sensitive to the parameter range and cost representation. Equation (2.7) allows the user excellent control over the desired outcome. Picking a good *wt* is not always easy though.

2.2.2 Parameter Representation

The binary genetic algorithm works with a finite (but usually extremely large) parameter space. This characteristic makes the genetic algorithm ideal for optimizing a cost that is due to parameters that can only assume a finite number of values. Common examples include choosing values from a list or parts from stock on hand.

If a parameter is continuous, then it must be quantized. Quantizing a parameter or signal is an established art. First, the range is divided into equal quantization levels (Figure 2.6). Any value falling within one of the levels is set equal to the mid, high, or low value of that level. In general, setting the value to the mid value of the quantization level is best, because the largest error possible is half a level. Rounding the value to the low or high value of the level allows a maximum error equal to the quantization level. The mathematical formulas for the binary encoding and decoding of the *n*th parameter, p_n , are given by

encoding:

$$p_{norm} = \frac{p_n - p_{lo}}{p_{hi} - p_{lo}} \quad (2.8)$$

$$gene[m] = \text{round} \left\{ p_{norm} - 2^{-m} - \sum_{p=1}^{m-1} gene[p] 2^{-p} \right\} \quad (2.9)$$

$$\frac{1_{lo}}{M_{lo}} \quad (2.6)$$

um and minimum values,
the lo and hi subscripts
eters. If one portion of the
weighted ($0 < wt < 1$):

$$\frac{1 - M_{lo}}{hi - M_{lo}} \quad (2.7)$$

ter range and cost repre-
t control over the desired
ough.

e (but usually extremely
as the genetic algorithm
s that can only assume a
de choosing values from

quantized. Quantizing a
he range is divided into
falling within one of the
of that level. In general,
on level is best, because
g the value to the low or
qual to the quantization
coding and decoding of

$$(2.8)$$

$$\left. gene[m]2^{-p} \right\} \quad (2.9)$$

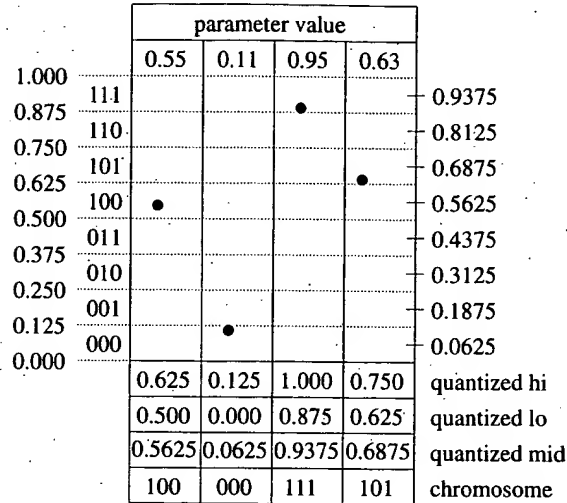


Figure 2.6 Four continuous parameter values are graphed with the quantization levels shown. The corresponding gene or chromosome indicates the quantization level where the parameter value falls. Each chromosome corresponds to a low, mid, or high value in the quantization level. Normally, the parameter is assigned the mid value of the quantization level.

decoding:

$$p_{quant} = \sum_{m=1}^{N_{gene}} gene[m]2^{-m} + 2^{-(M+1)} \quad (2.10)$$

$$q_n = p_{quant}(p_{hi} - p_{lo}) + p_{lo} \quad (2.11)$$

where

p_{norm} = normalized parameter, $0 \leq p_{norm} \leq 1$

p_{lo} = smallest parameter value

p_{hi} = highest parameter value

$gene[m]$ = binary version of p_n

$round\{\cdot\}$ = round to nearest integer

N_{gene} = number of bits in the gene

p_{quant} = quantized version of p_{norm}

q_n = quantized version of p_n

The genetic algorithm works with the binary encodings, but the cost function often requires continuous parameters. Whenever the cost function

is evaluated, the chromosome must first be decoded using equation (2.10). An example of a binary encoded chromosome that has N_{par} parameters, each encoded with $N_{gene} = 10$ bits, is

$$chromosome = \left[\underbrace{1111001001}_{gene_1} \underbrace{0011011111}_{gene_2} \dots \underbrace{0000101001}_{gene_{N_{par}}} \right]$$

Substituting each gene in this chromosome into equation (2.10) yields an array of the quantized version of the parameters. This chromosome has a total of $N_{bits} = N_{gene} \times N_{par} = 10 \times N_{par}$ bits.

As previously mentioned, the topographical map of Rocky Mountain National Park has 128×128 elevation points. If x and y are encoded in two genes, each with $N_{gene} = 7$ bits, then there are 2^7 possible values for x and y . These values range from $40^\circ 15' \leq y \leq 40^\circ 16'$ and $105^\circ 37'30'' \geq x \geq 105^\circ 36'$. Thus, a random chromosome may have the following $N_{bits} = 14$ -bit binary representation:

$$chromosome = \left[\underbrace{1100011001}_{x} \underbrace{1001}_{y} \right]$$

This chromosome translates into matrix coordinates of $[99, 25]$ or longitude, latitude coordinates of $[105^\circ 36'50'', 40^\circ 15'29.7'']$.

2.2.3 Initial Population

The genetic algorithm starts with a large commune of chromosomes known as the initial population. This initial population has N_{ipop} chromosomes and is an $N_{ipop} \times N_{bits}$ matrix filled with random ones and zeros generated from

$$IPOP = \text{round}\{\text{random}(N_{ipop}, N_{bits})\}$$

where the function $\text{random}(N_{ipop}, N_{bits})$ generates an $N_{ipop} \times N_{bits}$ matrix of uniform random numbers between zero and one. This type of function is available on all standard mathematical software. The function $\text{round}\{\}$ rounds the numbers to the closest integer. Each row in the matrix is a chromosome. The chromosomes correspond to discrete values of longitude and latitude. Next, the parameters are passed to the cost function for evaluation. A large initial population provides the genetic algorithm with a nice sampling of the search space. Usually, not all the initial population matrix chromosomes make the cut for the iterative portion of the genetic algorithm. Table 2.1 shows the initial population and their costs for the

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.